TABLE 9-continued

a	b	c	d	e
a > d	b > d	c > d	d > c	e > c
a < c	b < c	c < b	d < b	e < b
a < e	b < e	c < e	d < e	e < d
a > c	b > c	c >	d > b	e > b
a > e	b > e	c > e	d > e	e > d
a < b	b < a	c < a	d < a	e < a
a < d	b < d	c < d	d < c	e < c
a > b	b > a	c > a	d > a	e > a
a > e	b > e	c > e	d > e	e > d
a < c	b < c	c < b	d < b	e < b
a < d	b < d	c < d	d < c	e < c
a > c	b > c	c > b	d > b	e > b
a > d	b > d	c > d	d > c	e > c
a < b	b < a	c < a	d < a	e < a
a < e	b < e	c < e	d < e	e < d

After this initial assignment, the remaining relationships in each column (which are all duplicates of the relationships already assigned) are assigned to the registers. These new assignments are made, within each column, consistently with the assignments already made. Thus, a given inequality within a single column, such as "a>c", is always assigned to the same bit register: "a>c" is always assigned to R2 within column a. The complete table of assignment is shown in Table 10.

TABLE 10

	a	b	c	d	e
P1	a > b→R1	b > a→R1	c > a→R1	d > a→R1	e > a→R1
(R1&R2&	a > c→R2	b > c→R2	c > b→R2	d > b→R2	e > b→R2
R3&R4)	a < d→R3	b < d→R3	c < d→R3	d < c→R3	e < c→R3
	a < e→R4	b < e→R4	c < e→R4	d < e→R4	e < d→R4
P2	a > d→R5	b > d→R5	c > d→R5	d > c→R5	e > c→R5
(R5&R6&	a > e→R6	b > e→R6	c > e→R6	d > e→R6	e > d→R6
R7&R8)	a < b→R7	b < a→R7	c < a→R7	d < a→R7	e < a→R7
	a < c→R8	b < c→R8	c < b→R8	d < b→R8	e < b→R8
P3	a > b→R1	b > a→R1	c > a→R1	d > a→R1	e > a→Rl
(R1&R5&	a > d→R5	b > d→R5	c > d→R5	d > c→R5	e > c→R5
R8&R4)	a < c→R8	b < c→R8	c < b→R8	d < b→R8	e < b→R8
	a < e→R4	b < e→R4	c < e→R4	d < e→R4	e < d→R4
P4	a > c→R2	b > c→R2	c > b→R2	d > b→R2	e > b→R2
(R2&R6&	a > e→R6	b > e→R6	c > e→R6	d > e→R6	e > d→R6
R7&R3)	a < b→R7	b < a→R7	c < a→R7	d < a→R7	e < a→R7
	a < d→R3	b < d→R3	c < d→R3	d < c→R3	e < c→R3
P5	a > b→R1	b > a→R1	c > a→R1	d > a→R1	e > a→R1
(R1&R6&	a > e→R6	b > e→R6	c > e→R6	d > e→R6	e > d→R6
R8&R3)	a < c→R8	b < c→R8	c < b→R8	d < b→R8	e < b→R8
	a < d→R3	b < d→R3	c < d→R3	d < c→R3	e < c→R3
P6	a > c→R2	b > c→R2	c > b→R2	d > b→R2	e > b→R2
(R2&R5&	a > d→R5	b > d→R5	c > d→R5	d > c→R5	e > c→R5
R7&R4)	a < b→R7	b < a→R7	c < a→R7	d < a→R7	e < a→R7
	a < e→R4	b < e→R4	c < e→R4	d < e→R4	e < d→R4

A left column has been added to Table 10 indicating that each row of this table corresponds to a parallel condition register. A given parallel condition register is calculated by ANDing the registers indicated in the corresponding row of the table as follows:

(19) P1=R1 AND R2 AND R3 AND R4

(20) P2=R5 AND R6 AND R7 AND R8

(21) P3=R1 AND R5 AND R8 AND R4

(22) P4=R2 AND R6 AND R7 AND R3

(23) P5=R1 AND R6 AND R8 AND R3

(24) P6=R2 AND R5 AND R7 AND R4

A relationship table is compiled based on the assignments of Table 10. Within the first column, Table 10 indicates that "a>b" is assigned to R1. "a>c" is assigned to R2, and so on. Table 11 illustrates the resulting relationship table.

TABLE 11

	a	b	c	d	e
R1	a > b	b > a	c > a	d > a	e > a
R2	a > c	b > c	c > b	d > b	e > b
R3	a < d	b < d	c < d	d < c	e < c
R4	a < e	b < e	c < e	d < e	e < d
R5	a > d	b > d	c > d	d > c	e > c
R6	a > e	b > e	c > e	d > e	e > d
R7	a < b	b < a	c < a	d < a	e < a
R8	a < c	b < c	c < b	d < b	e < b

The relationship table is filled in accordance with the actual values of a, b, c, d, and e. Then, the parallel condition registers are calculated using equations 19–24. The result vector V1 is calculated as follows:

(25) V1=P1 OR P2 OR P3 OR P4 OR P5 OR P6

As already explained, V1 can be examined to find the median value of the set. At least one bit of V1 will be set. The bit position of this bit indicates which of the pixel values is the median.

To find the actual value of the median pixel, it is sometimes necessary to refer to some other data structure. For example, it might be necessary to refer to an actual pixel storage location in video memory. Vector V1 can be used as an index to a lookup table that contains the actual value. Alternatively, the indexed lookup table might contain a reference to a memory location in video memory containing the actual value. In the embodiment described herein, the lookup table contains an offset for each possible value of V1. Each offset indicates the addressing distance, in video memory, from the current pixel to the pixel that has been found to be the median.

It is possible to slightly optimize the procedure given above by eliminating one of the columns from the relationship table. For example, column e could be eliminated from Table 11. If e turns out to be the median value, this can be inferred from the absence of any true values in the result vector-the absence of such values will imply that column e is the median.

The invention is useful in any situation in which it is desired to find a median value from a set of given values. However, the invention is particularly efficient in environments such as video, graphics, and other signal processing environments where it is often desired to calculate median values for a large number of value sets. In this environment, several sets can be processed in parallel using SIMD instructions, thereby greatly increasing the efficiency of the calculations.

Although the invention has been described in language specific to structural features and/or methodological steps, it is to be understood that the invention defined in the appended claims is not necessarily limited to the specific features or steps described. Rather, the specific features and steps are disclosed as preferred forms of implementing the claimed invention.

What is claimed is:

1. A method of filtering pixels in a graphics image, wherein the pixels have pixel values, comprising the following steps:

defining different sets of pixel values, each set comprising the value of a center pixel and the values of at least some the pixels adjacent to the center pixel;

forming a plurality of bit registers having bit groups corresponding respectively to the different sets of pixel values, each bit group having bit positions correspond-